

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-227906

(43)Date of publication of application : 15.08.2000

(51)Int.Cl. G06F 15/16
G06F 9/46

(21)Application number : 11-335166

(71)Applicant : HITACHI LTD

(22)Date of filing : 25.11.1999

(72)Inventor : SATO YOSHIRO
MATSUMURA EIJI

(30)Priority

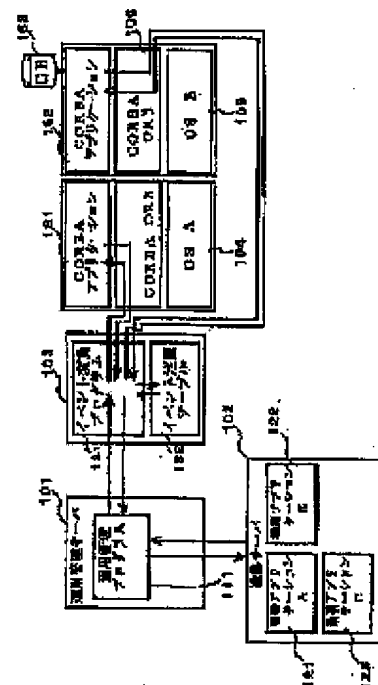
Priority number : 10338555 Priority date : 30.11.1998 Priority country : JP

(54) DISPERSED APPLICATION COOPERATION METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To cooperatively operate a program by converting an event which does not correspond to a dispersed object specification into an event corresponding to the dispersed object specification, issuing and transferring the event to an object in a dispersed object environment.

SOLUTION: An operation management program 111 previously issues an event which does not correspond to a dispersed object specification from a program which does not correspond to the dispersed object specification and a program which does not correspond to an object specification in a dispersed object environment. The event conversion program 131 of a server 103 converts the event which does not correspond to the dispersed object specification into the event corresponding to the dispersed object specification, issues and transfers the event to an object in the dispersed object environment.



LEGAL STATUS

[Date of request for examination] 03.02.2003

[Date of sending the examiner's decision of rejection] 05.11.2004

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-227906
(P2000-227906A)

(43) 公開日 平成12年8月15日 (2000.8.15)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード (参考)
G 0 6 F 15/16	6 2 0	G 0 6 F 15/16	6 2 0 T
9/46	3 6 0	9/46	3 6 0 F

審査請求 未請求 請求項の数 4 O L (全 9 頁)

(21) 出願番号 特願平11-335166

(22) 出願日 平成11年11月25日 (1999.11.25)

(31) 優先権主張番号 特願平10-338555

(32) 優先日 平成10年11月30日 (1998.11.30)

(33) 優先権主張国 日本 (J P)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 佐藤 善郎

神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア開発本部内

(72) 発明者 松村 榮二

神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア開発本部内

(74) 代理人 100096954

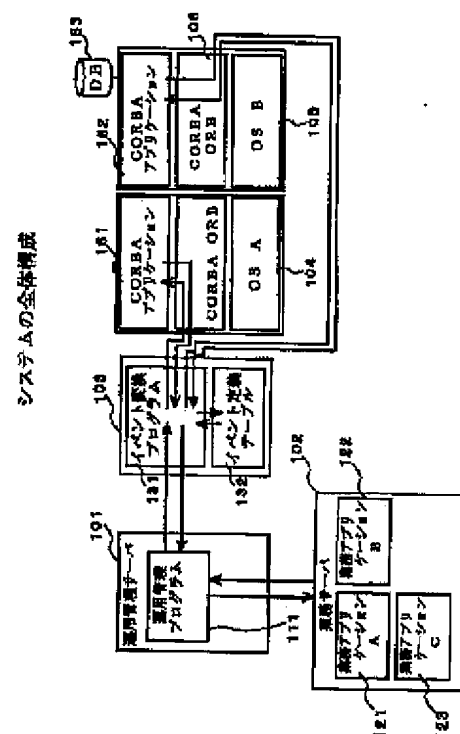
弁理士 矢島 保夫

(54) 【発明の名称】 分散アプリケーション連携方法

(57) 【要約】

【課題】 分散オブジェクト環境に対応していないプログラムと分散オブジェクト環境に対応したオブジェクトとを連携させて動作できるようにした分散アプリケーション連携方法を提供することを目的とする。

【解決手段】 分散オブジェクト仕様に対応していないイベントと分散オブジェクト仕様に対応したイベントとの間の変換を行なって、一方の環境下で発行されたイベントを他方の環境下のイベントに変換して発行するようにする。また分散オブジェクト環境下のオブジェクトに関する情報を収集するオブジェクトを用意し、該オブジェクトで分散オブジェクト環境下のオブジェクトに関する情報を収集し、分散オブジェクト対応でないプログラムに渡す。



【特許請求の範囲】

【請求項1】分散オブジェクト仕様に对应していないプログラムと分散オブジェクト環境下のオブジェクトとを連携して動作させる連携方法であって、分散オブジェクト仕様に对应していないプログラムから、分散オブジェクト仕様に对应していないイベントを発行するステップと、

該分散オブジェクト仕様に对应していないイベントを、分散オブジェクト仕様に对应したイベントに変換して発行するステップと、

該分散オブジェクト仕様に对应したイベントを、分散オブジェクト環境下のオブジェクトに渡すステップとを備えたことを特徴とする分散アプリケーション連携方法。

【請求項2】分散オブジェクト仕様に对应していないプログラムと分散オブジェクト環境下のオブジェクトとを連携して動作させる連携方法であって、分散オブジェクト環境下のオブジェクトから、分散オブジェクト仕様に对应したイベントを発行するステップと、

該分散オブジェクト仕様に对应したイベントを、分散オブジェクト仕様に对应していないイベントに変換して発行するステップと、

該分散オブジェクト仕様に对应していないイベントを、分散オブジェクト仕様に对应していないプログラムに渡すステップとを備えたことを特徴とする分散アプリケーション連携方法。

【請求項3】請求項1または2の何れか1つに記載の分散アプリケーション連携方法であって、前記イベントを変換して発行するステップは、分散オブジェクト仕様に对应していないイベントと分散オブジェクト仕様に对应したイベントとの対応関係を記載したイベント定義テーブルを参照して行なうことを特徴とする分散アプリケーション連携方法。

【請求項4】請求項1から3の何れか1つに記載の分散アプリケーション連携方法であって、分散オブジェクト環境下の任意のオブジェクトに関する情報を収集するための情報収集オブジェクトを用意し、該情報収集オブジェクトにより分散オブジェクト環境下の任意のオブジェクトに関する情報を収集するステップと、

該情報収集オブジェクトにより収集した情報を、分散オブジェクト仕様に对应していないプログラムに渡すステップとを備えたことを特徴とする分散アプリケーション連携方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、分散オブジェクト環境に对应していないプログラムと分散オブジェクト環境に对应したアプリケーションとを連携させる方法に関する。

【0002】

【従来の技術】分散オブジェクトシステムは、ネットワーク上に分散配置されたオブジェクトが相互に連携しながら全体として機能するシステムのことである。オブジェクトとは、データと処理手続き（メソッド）とが一体となったプログラムのことである。分散オブジェクトシステムを実現するオブジェクト技術に関する仕様として、例えば、OMG（The Object Management Group）が策定したCORBA（Common Object Request Broker Architecture）や、Microsoft社が策定したDCOM（Distributed Component Object Model）がある。OMGは米国Object Management Group, Incの米国およびその他の国における商標、CORBAは米国Object Management Group, Incの米国およびその他の国における登録商標、Microsoftは米国およびその他の国における米国Microsoft Corp.の登録商標である。CORBAでは、オブジェクト同士の通信路となるオブジェクト・リクエスト・ブローカ（ORB）を経由して各種の機能を備えるオブジェクトが通信することにより全体として機能する。各オブジェクトが、ネットワーク上のどのマシンに実装されているかなどは意識する必要がないので、CORBAのインターフェース仕様を守る限り、個々のオブジェクト単位で自由に機能拡張や変更、インプリメンテーションを行なうことができる。オブジェクト指向技術を用いて、オブジェクト相互がコミュニケーションする技術として、特開平6-332870号が開示されている。

【0003】一方、従来より、CORBAのような分散オブジェクト技術を用いないシステム（以下、クライアント・サーバ環境と記す。）も多く開発され現実に使用されている。例えば、各種の業務処理を行なう業務アプリケーションが使用されている。業務アプリケーションでは、幾つかのアプリケーションを連携して実行するスケジュール機能が実現されている。スケジュール機能は、例えば、時刻Xに業務アプリケーションAを起動し、該業務アプリケーションAが正常終了したら業務アプリケーションBを起動し、該業務アプリケーションAでエラーが発生したら業務アプリケーションCを起動する、というように所定のスケジュールに沿って業務アプリケーションを実行していく機能である。

【0004】

【発明が解決しようとする課題】ところで、分散オブジェクト技術を用いないシステムから、分散オブジェクト仕様に沿ったアプリケーション（オブジェクト）を呼び出してその機能を利用することは、従来の技術ではできない。分散オブジェクト環境下のオブジェクトの機能を利用するためには、分散オブジェクト技術を用いないシステムを、分散オブジェクト仕様に合わせて改変する必要がある。例えば、CORBAやDCOMにしたがったオブジェクトの機能を利用したいときには、呼び出す側

にCORBAやDCOMの仕様にしがったインターフェースを作成する必要がある。

【0005】しかし、分散オブジェクト技術を用いないシステムを、分散オブジェクト仕様に改変するには、時間と費用がかかる。分散オブジェクト技術を用いないシステムでも、例えばクライアント・サーバ方式で各種の機能を実現する複数の業務アプリケーションを連携させて実行するものもあり、そのようなシステムでは全ての業務アプリケーションを分散オブジェクト環境に乗せるように改変するのは時間と費用がかかる。一方、従来実現されている機能の一部分を、分散オブジェクト仕様のオブジェクトで実行させようとしても、上述したように、分散オブジェクト仕様でないシステムから、分散オブジェクト環境下のオブジェクトを呼び出すことはできない。

【0006】本発明は、上述の従来技術の問題点に鑑み、分散オブジェクト環境に対応していないプログラムと分散オブジェクト環境に対応したオブジェクトとを連携させて動作できるようにした分散アプリケーション連携方法を提供することを目的とする。

【0007】

【課題を解決するための手段】上記目的を達成するため、本発明は、分散オブジェクト仕様に对应していないプログラムと分散オブジェクト環境下のオブジェクトとを連携して動作させる連携方法であって、分散オブジェクト仕様に对应していないプログラムから、分散オブジェクト仕様に对应していないイベントを発行するステップと、該分散オブジェクト仕様に对应していないイベントを、分散オブジェクト仕様に对应したイベントに変換して発行するステップと、該分散オブジェクト仕様に对应したイベントを、分散オブジェクト環境下のオブジェクトに渡すステップとを備えたことを特徴とする。

【0008】また、分散オブジェクト仕様に对应していないプログラムと分散オブジェクト環境下のオブジェクトとを連携して動作させる連携方法であって、分散オブジェクト環境下のオブジェクトから、分散オブジェクト仕様に对应したイベントを発行するステップと、該分散オブジェクト仕様に对应したイベントを、分散オブジェクト仕様に对应していないイベントに変換して発行するステップと、該分散オブジェクト仕様に对应していないイベントを、分散オブジェクト仕様に对应していないプログラムに渡すステップとを備えたことを特徴とする。

【0009】さらに、前記イベントを変換して発行するステップは、分散オブジェクト仕様に对应していないイベントと分散オブジェクト仕様に对应したイベントとの対応関係を記載したイベント定義テーブルを参照して行なうようにするとよい。分散オブジェクト環境下の任意のオブジェクトに関する情報を収集するための情報収集オブジェクトを用意し、該情報収集オブジェクトにより分散オブジェクト環境下の任意のオブジェクトに関する

情報を収集するステップと、該情報収集オブジェクトにより収集した情報を、分散オブジェクト仕様に对应していないプログラムに渡すステップとを、さらに備えるようにしてもよい。

【0010】

【発明の実施の形態】以下、図面を用いて本発明の実施の形態を説明する。

【0011】図1は、本発明に係る分散アプリケーション連携方法を適用したシステムの全体構成を示す。運用管理サーバ101には運用管理プログラム111が実装されており、これにより各種の業務アプリケーションを連携して実行して業務を遂行する。業務サーバ102には、各種の業務アプリケーション121、122、123が実装されている。運用管理プログラム111は、スケジュール機能を備えており、例えば時刻Xに業務アプリケーションAを起動し、該業務アプリケーションAが正常終了したら業務アプリケーションBを起動し、該業務アプリケーションAでエラーが発生したら業務アプリケーションCを起動する、というように、業務サーバ102内のアプリケーションをあらかじめ定められたスケジュールに沿って実行できる。

【0012】運用管理サーバ101および業務サーバ102は、一般的なクライアント・サーバ(C/S)環境下にあるものであり、CORBA仕様にしがった分散オブジェクト環境下のものではない。運用管理プログラム111によって起動された業務アプリケーションは実行結果を業務管理プログラムへ終了コードとして通知する。

【0013】一方、運用管理プログラム111から、分散オブジェクト環境下のプログラム(オブジェクト)を利用することができる。あらかじめ運用管理プログラム111では、分散オブジェクト環境のオブジェクトを呼び出すためのクライアント・サーバ環境イベントを登録しておく。運用管理プログラム111から該クライアント・サーバ環境イベントが発行されると、サーバ103のイベント変換プログラム131が、該クライアント・サーバ環境イベントを分散オブジェクト仕様に沿ったオブジェクトを起動するイベントに変換する。ここでは、分散オブジェクト環境としてCORBAを用いるので、上記分散オブジェクト仕様に沿ったオブジェクトを起動するイベントをCORBA環境イベントと呼ぶ。

【0014】イベント変換プログラム131で変換され発行されたCORBA環境イベントは、OS A104やOS B105などの各種のプラットフォーム(例えば、Windows(Windowsは米国およびその他の国における米国Microsoft Corp.の登録商標である)やUNIX(UNIXは、X/Open Company Limitedが独占的にライセンスしている米国ならびに他の国における登録商標である)などであるが、これらに限らず他のプラットフォームでもかまわない)上に実装されたCORBAのオブジェクト・

リクエスト・ブローカ(ORB)106を介して、所定のCORBAアプリケーション(CORBA仕様のオブジェクト)161に渡され、該CORBAアプリケーション161が起動する。

【0015】逆に、イベント変換機能を利用することによって、CORBAアプリケーション161からクライアント・サーバ環境下のプログラムにCORBA環境イベントを発行し、CORBA環境イベントを受け取ったクライアント・サーバ環境下のプログラムに処理を行わせることもできる。例えば、CORBAアプリケーション161の処理の結果をクライアント・サーバ環境下の運用管理プログラム111に返す場合などである。この場合、あらかじめCORBAアプリケーション161では、クライアント・サーバ環境下のプログラムに情報を送るためのイベント(CORBA環境イベント)を登録しておく。CORBAアプリケーション161から該CORBA環境イベントが発行されると、サーバ103のイベント変換プログラム131が、該CORBA環境イベントに対応するクライアント・サーバ環境イベントに変換する。イベント変換プログラム131で変換され発行されたクライアント・サーバ環境イベントは、運用管理プログラム111に渡され、これによりCORBAアプリケーション161からの情報を運用管理プログラム111に渡すことができる。

【0016】イベント変換プログラム131は、実際の構成では、運用管理サーバ101と同じサーバに置くこともできるし、また、分散オブジェクト環境のCORBAアプリケーション161が存在するサーバにおくこともできる。もちろんイベント変換専用のサーバに運用管理サーバと、CORBAアプリケーションの両方から独立しておくこともできる。この場合、イベント変換プログラム131が存在するサーバ上には、CORBAのオブジェクト・リクエスト・ブローカ(ORB)106が存在する必要がある。

【0017】図4は、イベント変換プログラム131がクライアント・サーバ環境イベントとCORBA環境イベントとの間の変換を行なう際に参照するイベント定義テーブル400(図1の132)を示す。イベント定義テーブル400は、イベント名称のフィールド401とイベントIDのフィールド402を有する。イベント名称のフィールド401には、CORBA環境イベントの名称を登録する。イベントIDのフィールド402には、フィールド401に登録されたCORBA環境イベントに対応するクライアント・サーバ環境イベントのID(識別子)を登録する。

【0018】なお、本実施例ではCORBA環境イベント名称とクライアント・サーバ環境イベントのイベントIDとを対応させるイベント定義テーブル400を例として示したが、CORBA環境イベントのイベント本体の情報やクライアント・サーバ環境イベントのイベント

本体の情報をイベント定義テーブルに追加したり、CORBA環境イベントとクライアント・サーバ環境イベントとのイベント本体の定義テーブルを別途設けることで、イベント本体の内容の変換も可能である。

【0019】図5は、CORBA環境イベントのデータ形式の例を示す。CORBA環境イベント500は、CORBA環境下で有意なイベント名称501、該イベント名称のCORBA環境イベントに対応するクライアント・サーバ環境イベントのイベントID502、および当該CORBA環境イベントの内容を表すイベント本体503を備えている。

【0020】図6は、クライアント・サーバ環境イベントのデータ形式の例を示す。クライアント・サーバ環境イベント600は、当該クライアント・サーバ環境イベントを特定する識別子であるイベントID601、および当該クライアント・サーバ環境イベントの内容を表すイベント本体602を備えている。

【0021】図2は、イベント変換プログラム131においてクライアント・サーバ環境イベントを受信しCORBA環境イベントに変換して発行する処理の手順を示す。ステップ201で、クライアント・サーバ環境下で発行されたクライアント・サーバ環境イベント(図6の形式)を受け取る。ステップ202で、受け取ったクライアント・サーバ環境イベント600のイベントID601が、図4に示したイベント定義テーブル400に登録されているか否かを検索する。検索の結果、イベント定義テーブル400に登録されているイベントIDであった場合はステップ203から204に進み、登録されていないイベントIDであった場合はそのまま処理を終了する。

【0022】ステップ204では、受け取ったクライアント・サーバ環境イベント600をCORBA環境イベント500(図5の形式)に変換し、該CORBA環境イベントを発行して、処理を終了する。CORBA環境イベントへの変換は、具体的には、図5のデータ形式のCORBA環境イベント500を作成する処理である。CORBA環境イベント500中、イベントID502には、受け取ったクライアント・サーバ環境イベント600のイベントID601をセットする。イベント名称501には、当該イベントID601に対応するイベント名称を、イベント定義テーブル400から読み出してセットする。

【0023】イベント本体503には、受け取ったクライアント・サーバ環境イベント600のイベント本体602をCORBA環境のイベント本体に変換してセットする。

【0024】図3は、イベント変換プログラム131においてCORBA環境イベントを受信しクライアント・サーバ環境イベントに変換して発行する処理の手順を示す。ステップ301で、CORBA環境下で発行された

10

20

30

40

50

CORBA環境イベント(図5の形式)を受け取る。

【0025】ステップ302で、受け取ったCORBA環境イベント500のイベント名称501が、図4に示したイベント定義テーブル400に登録されているかどうか検索する。検索の結果、イベント定義テーブル400に登録されているイベント名称であった場合はステップ303から304に進み、登録されていないイベント名称であった場合はそのまま処理を終了する。

【0026】ステップ304では、受け取ったCORBA環境イベント500をクライアント・サーバ環境イベント600(図6の形式)に変換し、該クライアント・サーバ環境イベントを発行して、処理を終了する。クライアント・サーバ環境イベントへの変換は、具体的には、図6のデータ形式のクライアント・サーバ環境イベント600を作成する処理である。クライアント・サーバ環境イベント600中、イベントID501には、受け取ったCORBA環境イベント500のイベント名称501に対応するイベントIDを、イベント定義テーブル400から読み出してセットする。なお、このとき、イベント定義テーブル400から読み出したイベントIDが、受け取ったCORBA環境イベント500中のイベントID502に一致することをチェックする。イベント本体602には、受け取ったCORBA環境イベント500のイベント本体503をクライアント・サーバ環境のイベント本体に変換してセットする。

【0027】なお、図1では、クライアント・サーバ環境下のプログラムとして1つの運用管理プログラムを、分散オブジェクト環境下のプログラムとして1つのCORBAアプリケーション161を、それぞれ取り上げて説明したが、各環境下のプログラムは1つに限らず任意の数でよい。また、分散オブジェクト環境はCORBAに限らず、他の分散オブジェクト環境でもよい。クライアント・サーバ環境下の複数のプログラムと分散オブジェクト環境下の複数のプログラムとの間でも、上述したイベント変換プログラムによりイベント変換することにより、クライアント・サーバ環境下のプログラムから分散オブジェクト環境下のオブジェクトを呼び出してその機能を利用し、逆に分散オブジェクト環境下のオブジェクトからのアクションをクライアント・サーバ環境下のプログラムで受け取ることができる。したがって、クライアント・サーバ環境下のプログラムと分散オブジェクト環境下のオブジェクトとを連携させて動作させることができる。

【0028】次に、図1のシステムにおいて、クライアント・サーバ環境下の運用管理プログラム111でCORBAアプリケーション(オブジェクト)の各種の情報を収集する方法について説明する。まず、あらかじめ情報を収集したいCORBAアプリケーションに情報送信用ルーチンを用意しておく。この情報送信用ルーチンはオブジェクトとして構成されており、情報送信オブジェ

クトと呼ぶものとする。ここでは、図1のCORBAアプリケーション161の情報を収集する例で説明するので、該CORBAアプリケーション161内に情報送信オブジェクトが用意される。

【0029】CORBAアプリケーション162は、情報収集を実行するCORBAアプリケーション(オブジェクト)であり、データベース(DB)163を備えている。以下では、CORBAアプリケーション162を情報収集アプリケーション162と呼ぶ。情報収集アプリケーション162には、情報を収集したいCORBAアプリケーション161の情報送信オブジェクトを登録しておく。情報収集アプリケーション162は、所定時間間隔で、登録されている情報送信オブジェクトを呼び出す。呼び出された情報送信オブジェクトは、対応するCORBAアプリケーションの状態などの各種の情報を情報収集アプリケーション162に渡す。例えば、CORBAアプリケーション161の情報送信オブジェクトは、CORBAアプリケーション161に関する各種の情報を収集し情報収集アプリケーション162に渡す。情報収集アプリケーション162は、収集した各CORBAアプリケーションに関する情報をDB163に格納する。

【0030】情報収集アプリケーション162には、収集した情報をクライアント・サーバ環境下のプログラムに渡すためのオブジェクト(情報送信用オブジェクト)が用意されている。

【0031】運用管理プログラム111は、上述したイベント変換プログラム131を経由する方法で、上記情報送信用オブジェクトを呼び出すことができる。情報送信用オブジェクトは、DB163内に収集した各CORBAアプリケーションに関する情報を、上述したイベント変換プログラム131を経由する方法で、運用管理プログラム111に渡すことができる。

【0032】以上より、運用管理プログラム111は、任意のタイミングあるいはインターバルでDB163の情報(すなわちCORBAアプリケーションの状態など)を取得できる。したがって、CORBAアプリケーションに関する統計情報などをクライアント・サーバ環境の側で表示したり、取得した情報に応じた動作を行なうことなどが可能になる。

【0033】特に、従来の業務アプリケーションの情報収集の方法は、各機能を利用する側のタイミングで直接情報を収集するものであった。しかし、情報収集して管理する側のアプリケーションが複数ある場合は、同じ情報を何度も収集することになり、被管理アプリケーションへの負担(オーバーヘッド)が大きくなってしまう。

【0034】そこで、上述の本実施の形態のようにすれば、情報収集は情報収集アプリケーション162がまとめて行なうので、同じ情報を何度も収集することがなく、CORBA環境の被管理アプリケーションの負担は

少ない。クライアント・サーバ環境下の複数の管理プログラムから情報収集の要求があったときは、情報収集アプリケーション162がDB163に収集した情報をそれぞれ渡せばよいので、情報収集のためのオーバヘッドは小さく、クライアント・サーバ環境下の管理プログラムが増加しても、それほどの負担増にならない。

【0035】図7は、本発明を適用した障害監視システムの例を示す。監視サーバ701はクライアント・サーバ環境下にあり、運用管理プログラム702(図1の111と同様のもの)が動作している。運用管理プログラム702は、逆転シナリオ703にしたがうスケジュールで、分散オブジェクト環境下の業務アプリケーションあるいは不図示のクライアント・サーバ環境下の業務アプリケーションを動作させる。監視サーバ701は、業務アプリケーションで発生する障害を検出し、その障害発生に係る情報を障害ログ704に格納する。

【0036】サーバ711、業務サーバ721、およびWWWサーバ731は、分散オブジェクト(CORBA)環境にある。サーバ711では、イベント変換プログラム712が動作している。イベント変換プログラム712は、図1の131と同様のものである。業務サーバ721は、各種の業務アプリケーション(CORBAオブジェクト)を実装している。WWWサーバ731は、クライアント732〜734にWWWサービスを提供するサーバであり、必要に応じて業務サーバ721のオブジェクトを利用できる。

【0037】このようなシステムで、業務サーバ721に何らかの障害が発生したとき、その障害を通知するCORBA環境イベントが①に示すように発生される。このCORBA環境イベントは、図1で説明したようにイベント変換プログラム712によりクライアント・サーバ環境イベントに変換され、②に示すように監視サーバ701の運用管理プログラムに渡される。運用管理プログラム702は、当該障害発生に係る情報を障害ログ704に記録するとともに、サービス停止イベント(クライアント・サーバ環境イベント)を③に示すように発行する。このサービス停止イベントは、図1で説明したようにイベント変換プログラム712によりCORBA環境イベントのサービス停止イベントに変換され、④に示すように例えばWWWサーバ731に送られ、必要な停止処理が実行される。

【0038】図8は、本発明を適用したデータ収集システムの例を示す。運用管理サーバ801は、会社の管理部門に配置されたクライアント・サーバ環境下の管理用サーバである。この会社では、A店、B店、およびC店の3つの店舗を有し、各店舗にはそれぞれ店舗システム810、820、830が備えられている。店舗システム810は分散オブジェクト環境で構築され、店舗システム820は小規模なC/Sシステムで構築され、店舗システム830はメインフレームで構築されているもの

とする。

【0039】図8において、イベント変換プログラム811は運用管理サーバ801側に置くこともできるが、店舗810側に置くことによって、分散オブジェクト環境を局所化できる効果がある。すなわち、イベント変換プログラム811を運用管理サーバ801側に置いた場合は、運用管理サーバ801上にも分散オブジェクト環境を構築する必要があるが、店舗810側に置いた場合は、運用管理サーバ801側は従来のクライアント・サーバ環境のままで、店舗810にだけ分散オブジェクト環境を構築すれば良い。

【0040】運用管理サーバ801で各店舗システムのデータを収集したい場合、店舗システム810は分散オブジェクト環境であるので、上述した実施の形態と同様にイベント変換プログラム811を設けてデータ収集する。その手順は、運用管理サーバ801からデータ収集ジョブのクライアント・サーバ環境イベントを発行し、イベント変換プログラム811で該イベントを分散オブジェクト環境下のイベントに変換して発行し、店舗システム810からデータ収集の終了通知(分散オブジェクト環境下のイベント)が発行されたら、該終了通知のイベントをイベント変換プログラム811でクライアント・サーバ環境イベントに変換して運用管理サーバ801に渡す、という手順である。

【0041】さらに、分散オブジェクト環境ではない店舗システム820、830の場合は、例えば運用管理サーバ801からそれぞれの店舗システムに対して、直接ftpなどのファイル転送プログラムを呼び出すことができる。呼び出されたプログラムは、必要なデータを運用管理サーバ801に送る。送り終わったら、運用管理サーバ801にデータ転送作業完了を通知する。これらにより、運用管理サーバ801では、分散オブジェクト環境であるか否かにかかわらず、一元的にデータ収集することができる。

【0042】ウインドウ840は、運用管理サーバ801でデータ収集する際の手順定義画面の例を示す。なお、画面内の各部は本来はアイコンで表示されるものであるが、ここでは単なる四角いブロックで表した。各店舗システムからデータ収集し、それらをまとめてDBを更新し、そのDB情報に基づいて日報を出力したり業務障害通知を行なう手順が示されている。特に、A店のデータ収集では、分散オブジェクト環境のシステムからのデータ収集であるので、841に示すようにイベントの送受信でデータ収集することが示されている。

【0043】本実施形態においては、CORBA環境における一例を示したが、DCOMの環境においても本発明を実施することは可能である。

【0044】

【発明の効果】以上説明したように、本発明によれば、分散オブジェクト仕様にしたがっていないプログラムと

10

20

30

40

50

分散オブジェクト仕様にしたプログラムとの間で使用するイベントの変換を行なう。一方の環境下のプログラムで発行されたイベントを他方の環境下のプログラムのイベントに変換して発行するので、分散オブジェクト環境に対応していないプログラムから分散オブジェクト環境に対応したオブジェクト（プログラム）の機能を利用したり、分散オブジェクト環境に対応したオブジェクト（プログラム）からの情報を分散オブジェクト環境に対応していないプログラムに渡すなど、両環境でプログラムを連携させて動作させることができる。

【0045】特に、分散オブジェクト環境下のオブジェクトに関する情報を収集して、分散オブジェクト対応でないプログラムに渡せるので、分散オブジェクト対応でないプログラムから、分散オブジェクト環境のオブジェクト（プログラム）に関する情報を取得でき、該情報を表示したり管理に利用することができる。

【図面の簡単な説明】

【図1】分散アプリケーション連携方法を適用したシステムの全体構成図

【図2】クライアント・サーバ環境イベントをCORBA

A環境イベントに変換する処理のフローチャート図

【図3】CORBA環境イベントをクライアント・サーバ環境イベントに変換する処理を示すフローチャート図

【図4】イベント定義テーブルの例を示す図

【図5】CORBA環境イベントのデータ形式の例を示す図

【図6】クライアント・サーバ環境イベントのデータ形式の例を示す図

【図7】障害監視システムの例を示す図

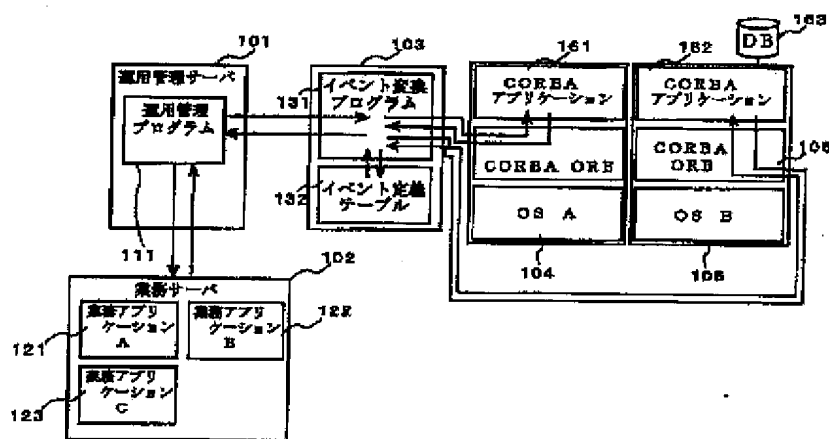
【図8】情報収集システムの例を示す図

【符号の説明】

101…運用管理サーバ、102…業務サーバ、103…サーバ、104、105…プラットフォーム、106…オブジェクト・リクエスト・ブローカ（ORB）、111…運用管理プログラム、121、122、123…業務アプリケーション、131…イベント変換プログラム、161、162…CORBAアプリケーション、163…データベース、400…イベント定義テーブル、500…CORBAイベントのデータ形式、600…C/S環境イベントのデータ形式。

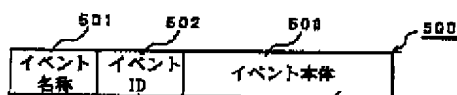
【図1】

システムの全体構成



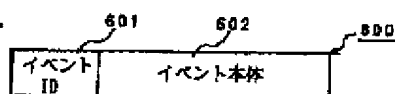
【図5】

CORBA環境イベントのデータ形式の例



【図6】

クライアント・サーバ環境(C/S環境)イベントのデータ形式の例



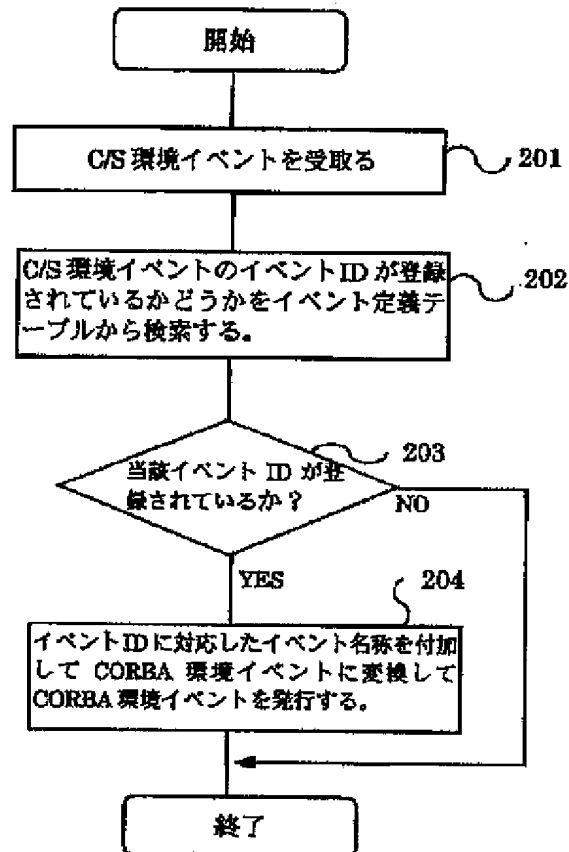
【図4】

イベント定義テーブルの例

イベント名称 401	イベントID 402	400
Event 1	E101	
Event 2	E102	
Event 3	E103	
Event 4	E104	

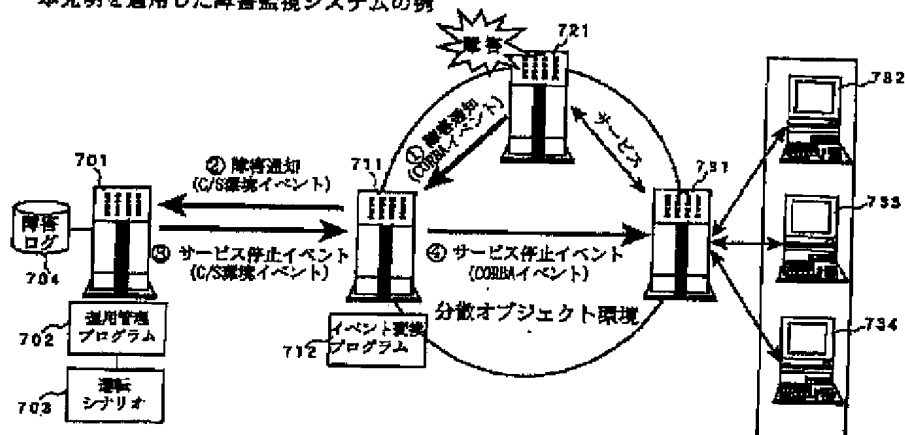
【 図2 】

クライアント・サーバ環境 (C/S 環境) イベントを
CORBA 環境イベントに変換する処理



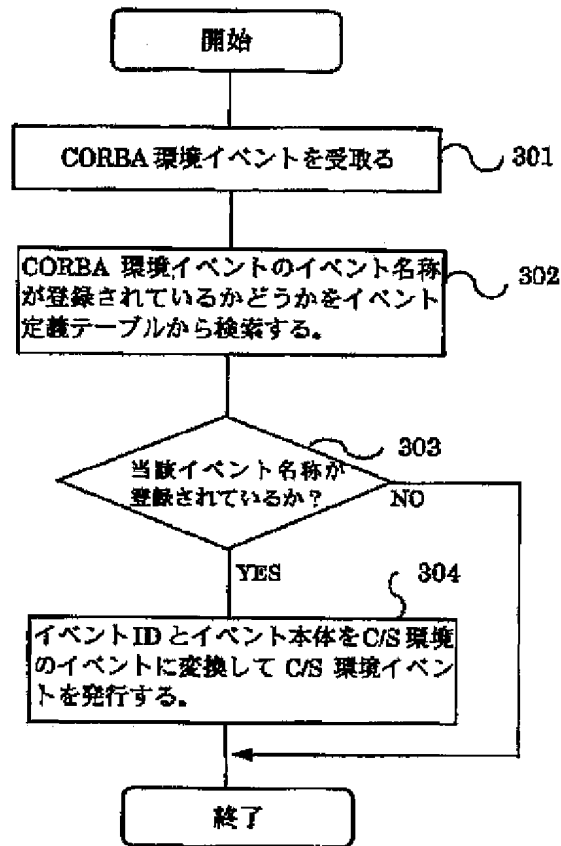
【 図7 】

本発明を適用した障害監視システムの例



【 図3 】

CORBA 環境イベントをクライアント・サーバ環境 (C/S 環境) イベント
に変換する処理



【 図8 】

